Auto-Scale Factor Circuit Realisation for MIMO Hardware Simulator

Bachir Habib, Gheorghe Zaharia, Ghaïs El Zein

Circuits and Systems, 2013, *, **
doi:10.4236/cs.2013.***** Published Online ** 2013 (http://www.scirp.org/journal/cs)

Auto-Scale Factor Circuit Realisation for MIMO Hardware Simulator

Bachir Habib, Gheorghe Zaharia, Ghais El Zein
Institute of Electronics and Telecommunications of Rennes, IETR, UMR CNRS 6164, Rennes, France
Email: bachir.habib@insa-rennes.fr

Received April 22, 2013

Abstract

A hardware simulator reproduces the behavior of the radio propagation channel, thus making it possible to test "on table" the mobile radio equipments. The simulator can be used for LTE and WLAN 802.11ac applications, in indoor and outdoor environments. In this paper, the input signals parameters and the relative power of the impulse responses are related to the relative error and SNR of the output signals. After analyzing the influence of these parameters on the output error and SNR, an algorithm based on an Auto-Scale Factor (ASF) is analyzed in details to improve the precision of the output signals of the hardware simulator digital block architecture. Moreover, the circuit needed for the validation of this algorithm has been introduced, verified and realized. It is shown that this solution increases the output SNR if the relative powers of the impulse responses are attenuated. The new architecture of the digital block is presented and implemented on a Xilinx Virtex-IV FPGA. The occupation on the FPGA and the accuracy of the architecture are analyzed.

Keywords: Hardware simulator; FPGA; MIMO radio channel; 802.11ac; LTE;

1. Introduction

Tests of a radio communication system, conducted under actual conditions are difficult, because tests taking place on outdoor, for instance, are affected by random movements or even by the weather. Thus, to evaluate the performance of the recent communication systems, a channel hardware simulator is considered. With hardware simulators, it is possible to very freely simulate desired types of radio channels. Moreover, it provides the necessary processing speed and real time performance, as well as the possibility to repeat the tests for any Multiple-Input Multiple-Output (MIMO) system.

Over the past few decades, efforts have resulted in several designs and implementation of real time simulators. Early efforts were based on analog components [1-4]. The development of real time simulator starts in 1973 when in [1] they developed the first Rayleigh based channel simulator. The simulator used Zener diode to generate Gaussian random variable.

However, with the advent of digital computers, fast Analog-Digital Convertors (ADC) and Digital-Analog Convertors (DAC), the analog components were replaced by digital thereby increasing the reliability and flexibility of simulators. In [5], they first used discrete digital logic in its simulator. With the development of Digital Signal Processing (DSP), the DSP based simulators were developed. In [6], they used 16 bit fixed point DSP for implementation and simulated the Gaussian quadrature components along with the log-normally distributed Line-Of-Sight (LOS) component. In [7], they used DSP chips for the development of a narrow-band simulator. In [8], they reported a frequency selective simulator using DSP and integrated circuits. It had a baseband bandwidth of 10 MHz and maximum Doppler frequency of 100 Hz. In [9], they developed 6 taps wide-band channel simulator having maximum signal bandwidth of 20 MHz. It used two 32 bit DSP floating point processors. In [10], they used a hybrid DSP FPGA architecture to build a wide-band channel simulator. It was capable of simulating 12 delay taps and had a baseband bandwidth of 5 MHz. Satellite channel simulator has been developed in [11] using DSP platform. In [12], they developed a narrow-band fast and accurate simulator. In [13], they developed a 5 MHz 12 taps wide-band simulator using 12 DSPs (1 for each tap) for the generation of complex coefficients. A narrow-band DSP based channel simulator has also been developed in [14].

Over the last decade, the use of Field Programmable Gate Arrays (FPGA) in DSP applications has become quite common. With continuing increase of the FPGA

capacity, entire baseband systems can be mapped onto faster FPGAs for more efficient prototyping, testing and verification. Larger and faster FPGAs permit the integration of a channel simulator along with the receiver noise simulator and the signal processing blocks for rapid and cost-effective prototyping and design verification. As shown in [15], the FPGAs provide the greatest design flexibility and the visibility of resource utilization. Thus, the FPGA based simulators have also been developed and their implementations have been described [16-23].

Some hardware simulators are proposed by industrial companies like Spirent (VR5) [24] and Elektrobit (Propsim F8) [25]. The commercially available channel simulators may not offer the user enough flexibility when configuring the wireless channel parameters to test the system under different environmental conditions. Moreover, those simulators are often too expensive and therefore prohibitive for communications laboratory. A low cost channel simulator is therefore required that present different environments and provide the user flexibility to measure the performance of the wireless system under real environmental conditions.

MIMO technology has attracted attention in wireless communications, because it offers significant increases in data throughput and link range without additional bandwidth or increased transmit power. It achieves this goal by spreading the same total transmitter power over the antennas to achieve an array gain that improves the spectral efficiency (more bits per second per hertz of bandwidth) and/or to achieve a diversity gain that improves the link reliability (reduced fading). Because of these properties, MIMO is an important part of modern wireless communication standards such as Wireless Local Area Network (WLAN) 802.11ac and Long Term Evolution (LTE). Thus, a 2×2 MIMO channel is considered in this paper.

The objective of our work concerns the channel models and the digital block of the simulator. The design of the RF blocks was completed in a previous project [17, 26].

The channel models can be obtained from standard channel models, as the TGn IEEE 802.11n [27] and the 3GPP-LTE models [28], or from real measurements conducted with the MIMO channel sounder designed and realized at IETR [29]. In the MIMO context, little experimental results have been obtained regarding time-variations, partly due to limitations in channel sounding equipment [30]. However, theoretical models of impulse responses of time-varying channels can be obtained using Rayleigh fading [31, 32].

The MIMO hardware simulator realized at IETR is reconfigurable with sample frequencies not exceeding 200 MHz, which is the maximum value for FPGA Virtex-IV. The 802.11ac signal provides a sample frequency of 200 MHz. Thus, it is compatible with the FPGA Virtex-IV. However, in order to exceed 200 MHz for the sample frequency, more performing FPGA as Virtex-VII can be used [33]. At IETR, several architectures of the digital block of a hardware simulator have been studied, in both time and frequency domains [34, 35]. Typically, wireless channels are commonly simulated using finite impulse response (FIR) filters, as in [21, 36, 37]. The FIR filter output signal is a convolution between a channel impulse response and a fed signal in such a manner that the signal delayed by different delays is weighted by the channel coefficients, i.e. tap coefficients, and the weighted signal components are summed up. The channel coefficients are periodically modified to reflect the behavior of an actual channel. Nowadays, different approaches have been widely used in filtering, such as distributed arithmetic (DA) and canonical signed digits (CSDs) [20].

Using FIR filter in a channel simulator has however a limitation. With a FPGA Virtex-IV, it is impossible to implement a FIR filter with more than 192 multipliers (impulse response with more than 192 taps). To simulate an impulse response with more than 192 taps, the Fast Fourier Transform (FFT) module can be used. With a FPGA Vitrex-IV, the size N_F of the FFT module can reach 65536 samples. Thus, several frequency architectures have been considered and tested [20]. However, their disadvantages are high latency and high occupation on FPGA. Moreover, the considered frequency architectures operate correctly for signals not exceeding the FFT size. Thus, new frequency architecture avoiding this limitation has been presented and tested in [38]. In this paper, the number of taps is limited to 18 for each SISO channel, thus, to 18×4 taps for the 2×2 MIMO channel. Therefore, the time domain architecture is considered because the total number of taps does not exceed 192.

In this paper, the input signals parameters and the relative power of the impulse responses has been related to the relative error and SNR of the output signals. After analyzing the influence of these parameters on the output error and SNR, an improvement algorithm based on an Auto-Scale Factor (ASF) has been proposed and analyzed in details. In fact, to decrease the error at the simulator output signals, it is better to consider a large number of bits in the architecture for the input signal and for the impulse responses. In the context of mobile radio, the input signal and the impulse responses cannot be predicted and they can undergo fading and be strongly attenuated. If they are low, they will not be quantified on a sufficient number of bits. Thus, the error of the output

signals of the channel simulator will increase widely. The proposed solution consists on multiplying the input signal and the impulse responses by an ASF that increases the output signals and makes it possible to quantify them on a higher number of bits in order to decrease the error at the output. Moreover, the received signal is divided by the correct ASF to obtain the correct output. Thus, a circuit is introduced which control the input and output signals powers for each sample. The circuit are presented, designed and tested.

Tests are made with input signal that respects the bandwidth chosen between $[\Delta, B + \Delta]$ and by considering 2×2 MIMO architecture. In fact, the channel impulse responses can be presented in baseband with its complex values, or as real signals with limited bandwidth B between $f_c - B/2$ and $f_c + B/2$, where f_c is the carrier frequency. In this paper, to eliminate the complex multiplication and the f_c, the hardware simulation operates between Δ and $B + \Delta$, where Δ depends on the band-pass filters (RF and IF). The value Δ is introduced to prevent spectrum aliasing. In addition, the use of a real impulse response allows the reduction by 50% of the size of the FIR filters and by 4 the number of multipliers. Thus, within the same FPGA, larger MIMO channels can be simulated.

The rest of this paper is organized as follows. Section 2 presents the channel models used for the test. Section 3 describes the simple and the ASF-based time domain architecture of the simulator digital block. In Section 4, the accuracy of the output signals of the architecture are analyzed in term of occupation on the FPGA and precision of the output signals. Lastly, Section 5 gives concluding remarks and prospects.

2. Channel Description

A MIMO propagation channel is composed of several time variant correlated SISO channels. For MIMO 2×2 channel, the received signals $y_j(t,\tau)$ can be calculated using a convolution :

(1)

The associated spectrum is calculated by the Fourier transform (using FFT modules):

(2)

The development of the digital block of a channel hardware simulator requires a good knowledge of the propagation channel. The different models of channels presented in literature used to apprehend as faithfully as possible the behavior of the channel.

Two channel models are considered to cover indoor and outdoor environments: the TGn channel models (indoor) and the 3GPP-LTE channel models (outdoor). Moreover, using the channel sounder realized at IETR, measured impulse responses are obtained for specific environments: shipboard, outdoor-to-indoor.

2.1. TGn Channel Models

TGn channel models [27] have a set of 6 profiles, labeled A to F, which cover all the scenarios. Each model has a number of clusters. For example, model E has four clusters. Each cluster corresponds to specific tap delays, which overlaps each other in certain cases. Reference [27] summaries the relative power of the impulse responses for TGn channel model E by taking the LOS impulse response as reference. According to the standard (WLAN 802.11ac) and its bandwidth, the sampling frequency is f_s = 165 MHz and the sampling period is $T_s = 1/f_s$.

2.2. 3GPP-LTE Channel Models

3GPP-LTE channel models are used for mobile wireless applications. A set of 3 channel models is used to simulate the multipath fading propagation conditions. A detailed description is presented in [28]. For LTE signals, f_s = 50 MHz.

2.3. Time-Varying Channels

In this section, we present the method used to obtain a model of a time variant channel, using Rayleigh fading [39] and based on Kronecker model [40]. The Doppler frequency f_d is equal to:

— (3)

where c is the celerity and v is the environmental speed. We have chosen a refresh frequency $f_{ref} > 2.f_d$ to respect the Nyquist-Shannon sampling theorem.

For an indoor environment (TGn model E for example), at f_c = 5 GHz and v = 4 km/h, f_d = 18.51 Hz. Thus, we have chosen f_{ref} = 40 Hz. For an outdoor environment (3GPP-LTE model EVA for example), at f_c = 1.8 GHz and v = 80 km/h, f_d = 133.27 Hz. Thus, we have chosen f_{ref} = 300 Hz.

The MIMO channel matrix H can be characterized by two parameters:

1) The relative power P_c of constant channel components which corresponds to the LOS.
2) The relative power P_s of the channel scattering components which corresponds to the Non-Line-Of-Sight (NLOS).

The ratio P_c/P_s is called Ricean K-factor. Assuming that all the elements of the MIMO channel matrix H are Rice distributed, it can be expressed for each tap by:

(4)

where H_F and H_V are the constant and the scattered channel matrices respectively.

The total relative received power $P = P_c + P_s$. Therefore:

(5)

(6)

If we combine (5) and (6) in (4) we obtain:

(7)

To obtain a Rayleigh fading channel, K is equal to zero, so H can be written as:

(8)

P is derived from [27] or [28] for each tap of the considered impulse response. For 2 transmit and 2 receive antennas:

(9)

where X_{ij} (i-th receiving and j-th transmitting antenna) are correlated zero-mean, unit variance, complex Gaussian random variables as coefficients of the variable NLOS (Rayleigh) matrix H_V.

To obtain correlated X_{ij} elements, a product-based model is used [40]. This model assumes that the correlation coefficients are independently derived at each end of the link:

(10)

H_w is a matrix of independent zero means, unit variance, complex Gaussian random variables. R_r and R_t are the receive and transmit correlation matrices. They can be written by:

(11)

where is the correlation between channels at two receives antennas, but originating from the same transmit antenna (SIMO). In other words, it is the correlation between the received power of channels that have the same Angle of Departure (AoD). is the correlation coefficient between channels at two transmit antennas that have the same receive antenna (MISO).

The use of this model has two conditions:

1) The correlations between channels at two receive (resp. transmit) antennas are independent from the *Rx* (resp. *Tx*) antenna.
2) If s_1, s_2 are the cross-correlation between antennas of the same side of the link, then :
   - $s_1 =$ + .
   - $s_2 =$ + .

For the uniform linear array, the complex correlation coefficients and are expressed by :

(12)

where $D = 2\pi d/\lambda$, $d = 0.5\lambda$ is the distance between two successive antennas, λ is the wavelength and R_{xx} and R_{xy} are the real and imaginary parts of the cross-correlation function of the considered correlated angles:

(13)

(14)

The Power Angular Spectrum (PAS) closely matchs the Laplacian distribution [41, 42]:

(15)

where σ is the standard deviation of the PAS (which corresponds to the numerical value of AS).

3. Digital block architecture

In this section, the architecture of the digital block of the hardware simulator is presented. First, the simple time domain architecture is described, and then the ASF-based architecture is presented and analyzed.

3.1. Simple Time Domain Architecture

We simulate 2×2 MIMO channel. Therefore, four FIR filters are considered to present the four SISO channels. For each channel, the FIR width and the number of used multipliers are determined by the taps of each channel. 4 FIR filters with 18 multipliers each are considered. Figure 1 presents two SISO channels of the time domain architecture based on FIR 147 filter with 18 multipliers.

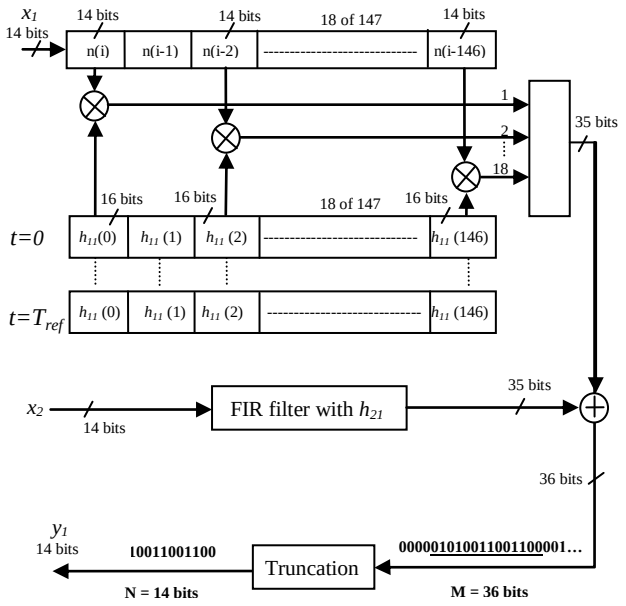

Figure 1. FIR 147 with 23 multipliers for one h_{11} and h_{21}

We have developed our own FIR filter instead of using Xilinx MAC FIR filter to make it possible to reload the FIR filter coefficients. The general formula for a FIR filter with 18 multipliers is:

In this relation, the index q suggests the use of quantified samples and $h_q(i_k)$ is the attenuation of the k^{th} path with the delay i_kT_s.

The truncation block is located at the output of the final digital adder. It is necessary to reduce the number of bits to 14 bits. Thus, these samples can be accepted by the digital-to-analog converter (DAC), while maintaining the highest accuracy. The immediate solution is to keep the first 14 bits. It is a "brutal" truncation (BT). This truncation decreases the real value of the quantified output sample. Moreover, 36 - 14 = 22 bits will be eliminated. Thus, instead of an output sample y, we obtain , where is the biggest integer number smaller or equal to u.

However, for low voltages, the brutal truncation generates zeros to the input of the DAC. Therefore, a better solution is the sliding truncation (ST) presented in Figure 1 which uses the 14 most significant bits. This solution modifies the output sample values. Therefore, the use of a reconfigurable amplifier after the DAC must be used to restore the correct output value. It must be divided by the corresponding sliding factor.

3.2. ASF-Based Time Domain Architecture

3.2.1. Why Using an ASF-Based Architecture?

To present the cause of using ASF-based time domain architecture, we related the input signals parameters and the relative power of the impulse responses to the relative error and SNR of the output signals. After analyzing the influence of these parameters on the output error and SNR, an improvement algorithm based on an Auto-Scale Factor ASF is proposed and analyzed in details.

In order to determine the accuracy of the digital block, a comparison is made between the theoretical/Xilinx output signals. An input Gaussian signal $x(t)$ is considered for the two inputs of the 2×2 MIMO simulator. To simplify the calculation, we consider $x(t) = x_1(t) = x_2(t)$:

$$x(t) = x_1(t) = x_2(t) = x_m e^{\frac{\left(t-m_x\right)^2}{2\sigma_x^2}},\ 0 \le t \le W_t \quad (17)$$

In fact, the FT of a Gaussian signal is also Gaussian signal, and to obtain a signal $x(t)$ that respect the bandwidth [, $+B$], the following steps are considered:

In frequency domain, the Gaussian input signal = FT is computed by:

$$X(f) = x_m\sigma_x\sqrt{2\pi}e^{-2\left(f\right)^2\sigma_x^2}e^{-j2\pi fm_x} \quad (18)$$

with

$$\left|X(f)\right| = x_m\sigma_x\sqrt{2\pi}e^{-2\left(f\right)^2\sigma_x^2} \quad (19)$$

This signal spectrum is limited between and $+B$ if:

$$6\sigma_X \le B \quad (20)$$

where is the standard deviation of . Comparing the first and the third equation, we obtain:

$$2\pi\sigma_X\sigma_x = 1 \quad (21)$$

Thus, that corresponds to the considered band of the standard used, is obtained:

$$\sigma_x \ge \frac{3}{\pi B} \quad (22)$$

To obtain $x(t)$ centered between [, $+B$], it must be multiplied by:

$$x(t) \to x(t).\cos\left(2\pi.\left(\frac{B}{2}+\Delta\right).t\right) \quad (23)$$

In our work, we considered $\sigma_x = 3/\pi B$. m_x is chosen equal to $20T_s > 3$ for both WLAN 802.11ac and LTE signals. Moreover, $<< B$ is chosen equal 2 MHz. These values are small enough to show the effect of each tap on the output signal. The ADC and DAC converters of the development board have a full scale $[-V_m, V_m]$, with V_m = 1 V. For the simulations, we consider $x_m = V_m/2$.

For WLAN 802.11ac, B = 80 MHz and $T_s = 1/f_s$ = 6 ns. Thus, we obtain $= 2T_s$. This signal is named $x_{WLAN}(t)$ and is presented in figure 2.

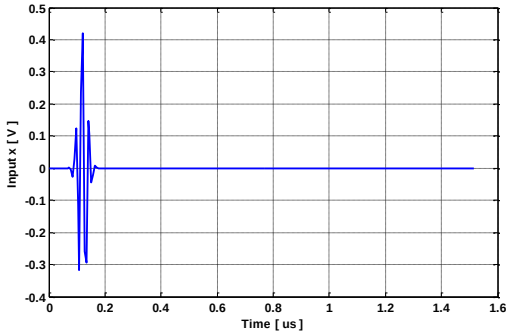

Figure 2. Input signal for WLAN 802.11ac for the 2×2 time domain architecture

For LTE, B = 20 MHz and $T_s = 1/f_s$ = 20 ns. Thus, we obtain $= 2.5T_s$. This signal is named $x_{LTE}(t)$ and is presented in figure 3.

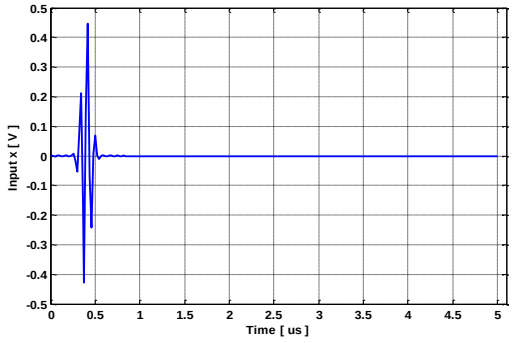

Figure 3. Input signal for LTE for the 2×2 time domain architecture

The global output SNR can be affected by the input signal and the impulse response. The output global SNR of the first output is calculated by:

$$RSNR_G(i) = 20\log_{10}\left\| \frac{y_1(i)}{y_{q1}(i) - y_1(i)} \right\| \quad \text{[illegible]} \tag{24}$$

$$= 20\log_{10}\left\| \frac{\sum_{k=1}^{9} \text{[illegible]}_{11}(i_k) + h_{21}(j_k) \text{[illegible]}(t - i_k T_s)}{\sum_{k=1}^{9} \text{[illegible]}_{q11}(i_k) + h_{q21}(j_k) \text{[illegible]}_q(t - i_k T_s) - \sum_{k=1}^{9} \text{[illegible]}_{11}(i_k) + h_{21}(j_k) \text{[illegible]}(t - i_k T_s)} \right\|$$

and for the second output by:

$$RSNR_G(i) = 20\log_{10}\left\| \frac{y_2(i)}{y_{q2}(i) - y_2(i)} \right\| \quad \text{[illegible]} \tag{25}$$

$$= 20\log_{10}\left\| \frac{\sum_{k=1}^{9} \text{[illegible]}_{12}(i_k) + h_{22}(j_k) \text{[illegible]}(t - i_k T_s)}{\sum_{k=1}^{9} \text{[illegible]}_{q12}(i_k) + h_{q22}(j_k) \text{[illegible]}_q(t - i_k T_s) - \sum_{k=1}^{9} \text{[illegible]}_{12}(i_k) + h_{22}(j_k) \text{[illegible]}(t - i_k T_s)} \right\|$$

where h_q and x_q are the quantified impulse responses and input signal respectively.

The parameters of the input signal that have an impact on the output global SNR are: x_m, , W_t and n_x, where n_x is the number of bits of the input signal. However, n_x is fixed by the ADC by 14 bits. Thus, it won't be considered in the study.

If x_m decreases, then decreases. Thus, x will be quantified on lower number of bits. Therefore, the global SNR decreases. Figure 4 presents the global output SNR versus x_m for the TGn model E and for 3GPP-LTE model EVA respectively.

If decreases, then more samples of the input signal x will be quantified on lower number of bits. Therefore, the global SNR decreases. The amount of the low values of x is related to and W_t.

Figure 5 presents the global output SNR versus for the TGn model E and for 3GPP-LTE model EVA respectively.

After a simple calculation, we notice that and W_t are linear related, as presented in figure 6.

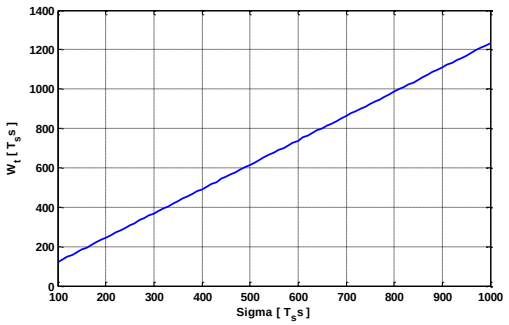

Figure 6. W_t versus for the input signal

Thus, we define a factor equal to the null part of x:

$$F_x = W_t - 6\sigma \tag{26}$$

If W_t increases or/and decreases, F_x increases which leads to a large interval of small values of x. Thus, the global SNR decreases. Figure 7 presents the global output SNR versus F_x.

The number of bits of the impulse response (h) has an impact on the output SNR. In fact, quantifying h on lower number of bits decreases the global output SNR. Figure 8 presents the output global SNR versus n_h, where n_h is the number of bits of h.

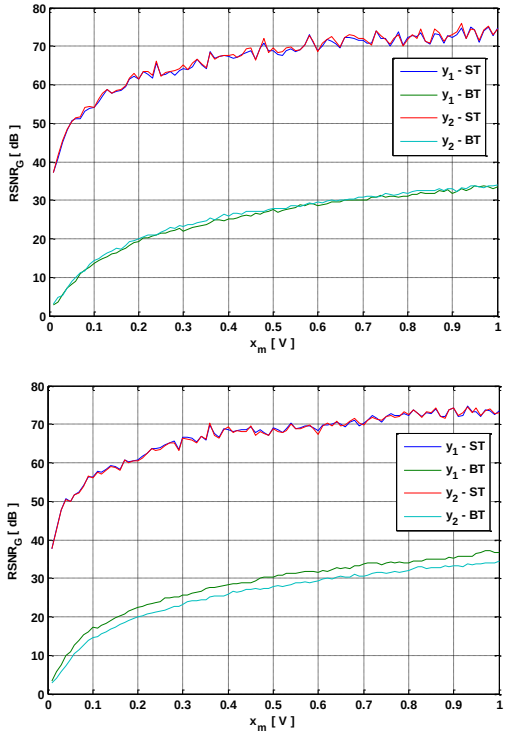

Figure 4. Global SNR versus x_m for TGn model E and 3GPP-LTE model EVA respectively

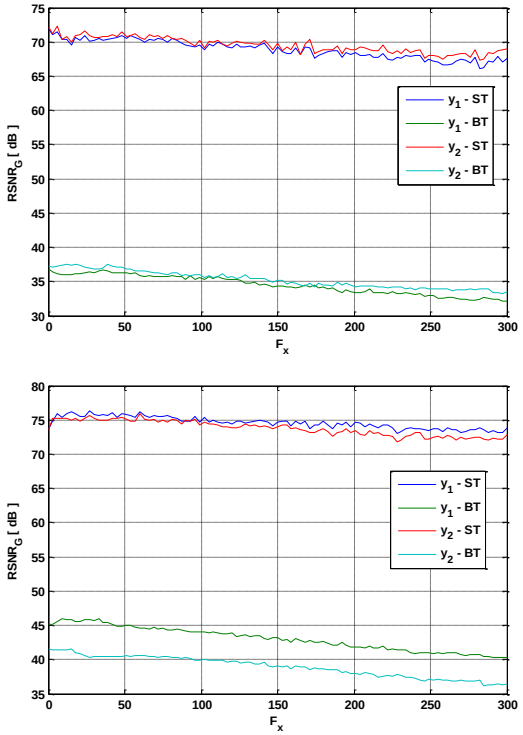

Figure 7. Global output SNR versus F_x for TGn model E and 3GPP-LTE model EVA respectively

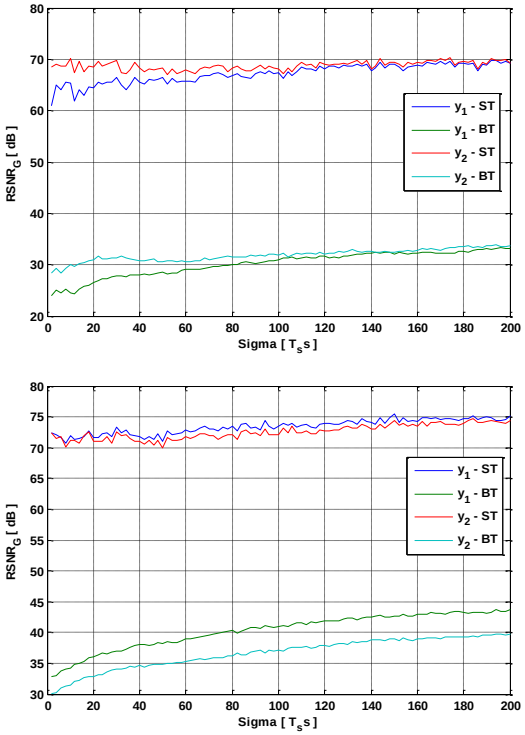

Figure 5. Global SNR versus for TGn model E and 3GPP-LTE model EVA respectively

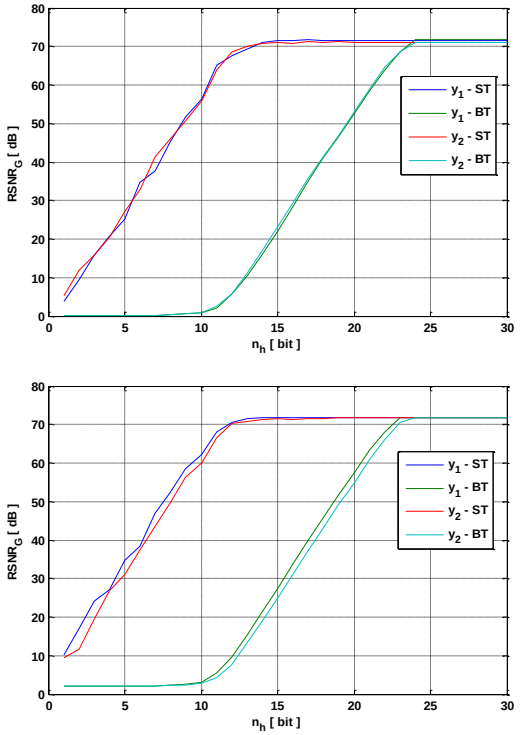

Figure 8. Global output SNR versus n_h for TGn model E and 3GPP-LTE model EVA respectively

3.2.2. ASF-Based Architecture Description

After analyzing the global relative SNR, we conclude that it is high only for high values of the input signals and the impulse response. Therefore, to decrease the error at the simulator output signals, it is better to consider a large number of bits in the architecture for the input signals and for the impulse response.

In the context of mobile radio, the input signals and the impulse response cannot be predicted and they can undergo fading and be strongly attenuated. If the signal is low, it will not be quantified on a sufficient number of bits. Thus, the error of the output signals of the channel simulator will increase widely. The proposed solution consists on multiplying the input signals and the impulse response by an ASF that increases the output signals and makes it possible to quantify them on a higher number of bits in order to decrease the error at the output. Moreover, the received signal is divided by the correct ASF to obtain the correct output.

Figure 9 presents the global ASF diagram. This diagram will be described and analyzed in details. The new 2×2 MIMO ASF-based architecture is presented in Figure 10.

The two signals $x_1(t)$ and $x_2(t)$ are the input signals of the 2×2 MIMO channel, and the two signals $y_1(t)$ and $y_2(t)$ are the output signals.

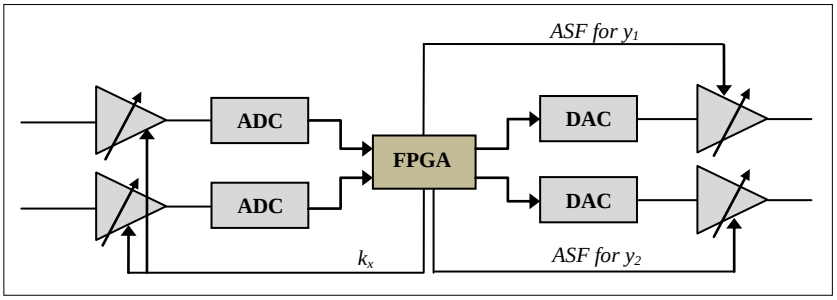

Figure 9. ASF diagram using the gain controlled amplifiers

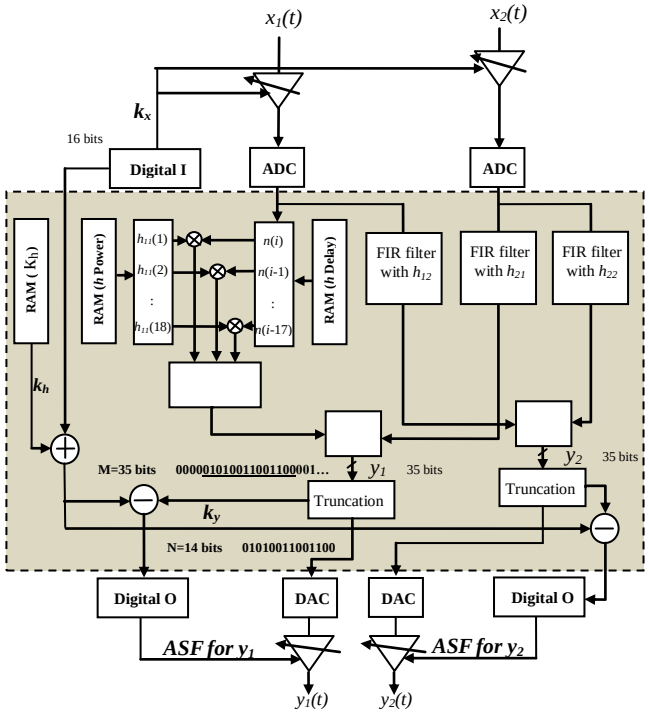

Figure 10. Principle scheme using ASF

The colored large block is the programmable digital part (Virtex-IV) of the hardware simulator. “I” stands for Input and “O” for Output.

The maximum voltage supported by the ADC is 1 V. If $x_1 < 0.25$ V, it is multiplied by where is the integer verifying:

$$\frac{1}{2} < 2^{k_{x_1}}.\ x_1 < 1 \tag{27}$$

with

$$k_{x_1} = \left\lceil \log_2\left(\frac{1}{x_1}\right)\right\rceil - 1 \tag{28}$$

In the same way, If x_2 < 0.25 V, it is multiplied by where is the integer verifying:

$$\frac{1}{2} < 2^{k_{x2}}.\ x_2 < 1 \tag{29}$$

with

$$k_{x_1} = \left\lceil \log_2\left(\frac{1}{x_2}\right)\right\rceil - 1 \tag{30}$$

The input signal cannot exceed 1 V. Thus, we define by:

$$k_x = \min\left\{k_{x_1}, k_{x_2}\right\} \tag{31}$$

In fact, we cannot provide the digital input by the signals and . Otherwise, the ASF calculated and provided to the digital output will be unknown.

If h_{max} = max (|h|) < 0.5, it is multiplied by where:

$$\frac{1}{2} < 2^{k_h}.\ h < 1 \tag{32}$$

with

$$k_h = \left\lceil \log_2\left(\frac{1}{h_{\max}}\right)\right\rceil - 1 \tag{33}$$

k_h is determined for every MIMO profile and it is saved in a RAM block in the FPGA. A gain controlled amplifier is placed before the ADC to control the input signals by before sending it to the FPGA. Also, controlling the power of the input signal at a sampling period smaller than the sampling period of the FPGA is not easy (or realistic). Thus, is provided for a package of input samples.

Figure 11 describes the electronic ship that we developed to control the input signals by . The figure present one control amplifier for one input signal. *U* is the initial power of the input signal and *R* are resistances.

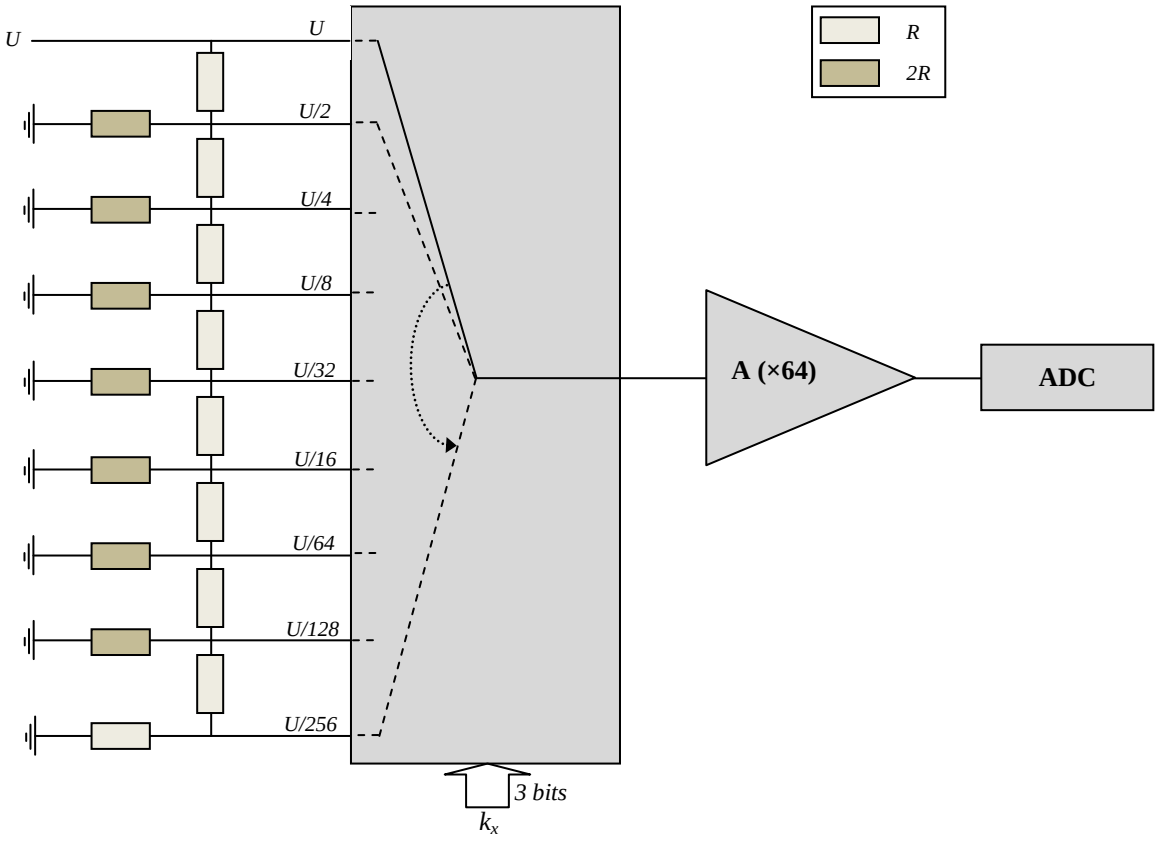

Figure 11. Electronic ship to control the input signal

In the case of a brutal truncation, *ASF* = + - where is the fixed brutal truncation equal to $2^{35-14} = 2^{21}$ (35 bits for the output before truncation and 14 bits for the output after truncation). Moreover, a better solution is the sliding truncation, described previously, that selects the most significant bits. The ADC and the DAC have a resolution of 14 bits. In this case, if the output signals are presented on more than 14 bits, the sliding factor has to be considered to obtain the correct output signal. Thus, *ASF* = , and using just the ASF on *h*, *ASF* = . The resulting *ASF* is sent to a gain controlled amplifier to restore the true value of the output signals, as presented in the figure of the global diagram of the ASF. The first MSB bit defines the multiplication or the division by 2^{ASF}.

A FPGA Virtex-IV provides 34 pins (digital I/O) Adjacent Bus Header on the motherboard of the FPGA. This will provides 28 direct bi-directional connections to the main user FPGA. The remainder of the pins provides a 3.3 V connection, a GND connection and they are No-Connects (NC). Thus, k_x provided to the input of the FPGA and the two *ASF* provided to the outputs y_1 and y_2 are quantified on 9 bits. In fact, as presented in the previous figure, 3 bits for the quantification is sufficient for this work. The input (output resp.) signal can be multiplied (divided resp.) by $2^3 = 256$ which is about 25 dB. Higher than 25 dB, the noise undergoes the *ASF* process and it will grow significantly.

4. Accuracy Evaluation

4.1. Occupation on FPGA

The Xtreme DSP Virtex-IV board from Xilinx [33] is used for the implementation. The XtremeDSP features dual-channel high performance ADCs (AD6645) and DACs (AD9772A) with 14-bit resolution, a user programmable Virtex-IV FPGA, programmable clocks, support for external clock, host interfacing PCI, two banks of ZBT-SRAM, and JTAG interfaces. The simulations and synthesis are made with Xilinx ISE [33] and ModelSim software [43].

The 2×2 MIMO architectures are implemented in the FPGA Virtex-IV which has 2 ADC and 2 DAC, it can be connected to only 2 down-conversion and 2 up conversion RF units. To test a higher order MIMO array, the use of more performing FPGA as Virtex-VII [33] is recommended.

Table 4 presents the FPGA utilization of 2×2 MIMO time domain architecture using four FIR filters with their additional circuits used to dynamically reload the channel coefficients.

In the FPGA, the clock is controlled by a Virtex-II which is connected to the Virtex-IV.

As the development board has 2 ADC and 2 DAC, it can be connected to only 2 down-conversion and 2 up-conversion RF units. Four FIR filters are needed to simulate a one-way 2×2 MIMO radio channel. The occupancy of the time domain architecture is known after performing operations of synthesis, mapping, place and route from the program written in VHDL. Table 1 shows the device utilization in one Virtex-IV SX35 for 2×2 MIMO channel using the time domain architecture for the TGn channel model E.

Table 1: Virtex-IV utilization summary for the 2×2 MIMO simple time domain architecture, for TGn model E

Logic Utilisation	Used	Available	Utilization
Slice Flip Flops	3,992	30,720	13 %
4 input LUTs	5,526	30,720	18 %
Occupied Slices	2,440	15,360	16 %
FIFO16/RAMB16s	1	192	1 %
DSP48s	72	192	38 %

Table 2 shows the device utilization in one Virtex-IV SX35 for 2×2 MIMO channel using the time domain architecture for the 3GPP-LTE model EVA.

Table 2: Virtex-IV utilization summary for the 2×2 MIMO simple time domain architecture, for 3GPP-LTE model EVA

Logic Utilisation	Used	Available	Utilization
Slice Flip Flops	3,296	30,720	11 %
4 input LUTs	4,097	30,720	14 %
Occupied Slices	1,891	15,360	13 %
FIFO16/RAMB16s	1	192	1 %
DSP48s	36	192	19 %

We notice that the occupation of slice on the FPGA of a 2×2 MIMO system is 16 % for the TGn channel model E and 16 % for the 3GPP-LTE model EVA. In fact, these occupations are equal to the occupations of a SISO channel multiplied by four and with additional slices added because of the two digital adders that operates $y_{11}+ y_{21}$ and $y_{12}+ y_{22}$. Moreover, the 2×2 MIMO system has small occupation on the FPGA Virtex-IV. In fact, we can implement up to 4×4 MIMO system in the FPGA for the 3GPP-LTE model EVA (because for TGn channel model E the number of multiplier is equal to 18×(4×4) = 288>192). However, we are limited by the 2 ADC and the 2 DAC.

Table 3 shows the device utilization in one Virtex-IV SX35, after performing operations of synthesis, mapping, place and route, for 2×2 MIMO channel using the ASF-based time domain architecture for the TGn channel model E.

Table 3: Virtex-IV utilization summary for the 2×2 MIMO ASF-based time domain architecture, for TGn model E

Logic Utilisation	Used	Available	Utilization
Slice Flip Flops	4,124	30,720	14 %
4 input LUTs	5,745	30,720	19 %
Occupied Slices	2,601	15,360	17 %
FIFO16/RAMB16s	2	192	2 %
DSP48s	72	192	38 %

Table 4 shows the device utilization in one Virtex-IV SX35 for 2×2 MIMO channel using the ASF-based time domain architecture for the 3GPP-LTE model EVA.

Table 4: Virtex-IV utilization summary for the 2×2 MIMO ASF-based time domain architecture, for 3GPP-LTE model EVA

Logic Utilisation	Used	Available	Utilization
Slice Flip Flops	3,425	30,720	12 %
4 input LUTs	4,347	30,720	15 %
Occupied Slices	2,074	15,360	14 %
FIFO16/RAMB16s	2	192	2 %
DSP48s	36	192	19 %

We notice that the occupation of slice on the FPGA using the ASF-based architecture increases about just 1 %. However, as we will see in the next section, the precision of the output signals increase significantly.

The channel impulse responses are stored on the hard disk of the computer and read via the PCI bus and then stored in the FPGA dual-port RAM. Figure 12 shows the connection between the computer and the FPGA board to reload the coefficients. The successive profiles are considered for the test of a 2×2 MIMO time-varying channel.

The maximum data transfer of the impulse responses is: 18 × 4 = 72 words of 16 bits = 162 bytes to transmit for a MIMO profile, which is: 162 × f_{ref} (Bps). f_{ref} depends on the environment.

The MIMO profiles are stored in a text file on the hard disk of a computer. This file is then read to load the memory block which will supply RAM blocks on the simulator (one block for each tap of the impulse response). Each block RAM has a memory of 64 (kB), thus 512 (kbits). The impulse responses are quantified on 16 bits, therefore, up to 32,000 MIMO profiles can be supplied in the RAM blocks. Each environment needs 4 blocks RAM for the power of the impulse responses and 4 blocks for the delays, which is a total of 8 blocks RAM. Reading the file can be done either from USB or PCI interfaces, both available on the used prototyping board. The PCI bus is chosen to load the profiles. It has a speed of 30 (MB/s). In addition, this is a bus of 32 (bits). Thus, on each clock pulse two samples of the impulse response are transmitted.

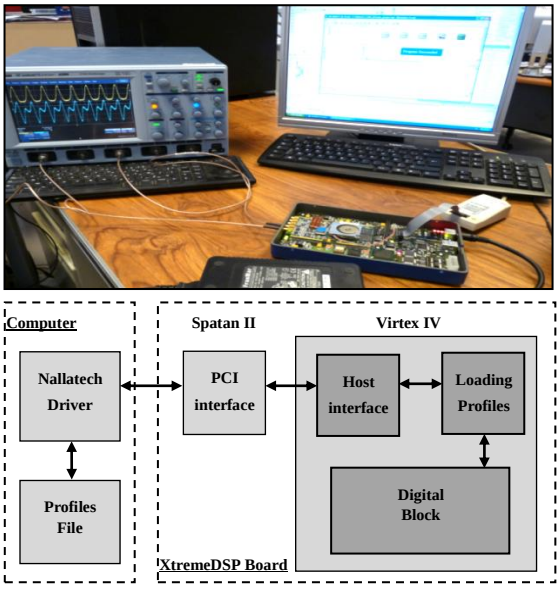

Figure 12. Connection between the computer and the XtremeDSP board

The Nallatech driver in Figure 12 provides an IP sent directly to the "Host Interface" that reads it from the PCI bus and stores these data in a FIFO memory. The module called "Loading profiles" reads and distributes the impulse responses in "RAM" blocks. While a MIMO profile is used, the following profile is loaded and will be used after the refresh period.

4.2. Output signal Precision

In order to determine the accuracy of the digital block, a comparison is made between the theoretical and the Xilinx output signals.

The theoretic output vector of the 2×2 MIMO channel is given previously. The relative error, which is given for each output sample, is calculated for the two outputs by:

$$RE(i) = \frac{y_{q1}(i) - y_1(i)}{y_1(i)}.100 \quad [\%] \quad \textit{for output 1} \quad (34)$$

$$RE(i) = \frac{y_{q2}(i) - y_2(i)}{y_2(i)}.100 \quad [\%] \quad \textit{for output 2} \quad (35)$$

where y_{q1} and y_{q2} is the vector containing the samples of the Xilinx output signals, and y_1 and y_2 are the theory output signals. $i = \overline{1, L + i_{Final}}$ and i_{Final} is computed by for the two outputs by:

$$i_{Final} = \max(i_{max11}, i_{max21}) \quad \text{for output 1} \quad (36)$$

$$i_{Final} = \max(i_{max12}, i_{max22}) \quad \text{for output 2} \quad (37)$$

where

i_{max11} = the index of the last tap $\neq 0$ of h_{11}
i_{max21} = the index of the last tap $\neq 0$ of h_{21}
i_{max12} = the index of the last tap $\neq 0$ of h_{12}
i_{max22} = the index of the last tap $\neq 0$ of h_{22}

The relative SNR is computed for the two outputs by:

$$RSNR(i) = 20\log_{10}\left|\frac{y_1(i)}{y_{q1}(i) - y_1(i)}\right| [dB] \quad \text{for output 1} \quad (38)$$

$$RSNR(i) = 20\log_{10}\left|\frac{y_2(i)}{y_{q2}(i) - y_2(i)}\right| [dB] \quad \text{for output 2} \quad (39)$$

The global values of the relative error and SNR computed for the output signals after the final truncations are necessary to evaluate the accuracy of the architecture. The global relative error is computed for the two outputs by:

$$RE_G = \frac{\|y_{q1}(i) - y_1(i)\|}{\|y_1(i)\|}.100 \quad [\%] \quad \text{for output 1} \quad (40)$$

$$RE_G = \frac{\|y_{q2}(i) - y_2(i)\|}{\|y_2(i)\|}.100 \quad [\%] \quad \text{for output 2} \quad (41)$$

The global SNR is computed by:

$$RSNR_G(i) = 20\log_{10}\left\|\frac{y_1(i)}{y_{q1}(i) - y_1(i)}\right\| [dB] \quad \text{for output 1} \quad (42)$$

$$RSNR_G(i) = 20\log_{10}\left\|\frac{y_2(i)}{y_{q2}(i) - y_2(i)}\right\| [dB] \quad \text{for output 2} \quad (43)$$

Figure 13 presents the effect of the *ASF* on the global output SNR versus the attenuation of the impulse response *h* using TGn channel model E. Figure 14 presents the variation of and versus the attenuation of *h* using TGn channel model E.

Figure 15 presents the effect of the *ASF* on the global output SNR versus the attenuation of *h* using 3GPP-LTE channel model EVA. Figure 16 presents the variation of and versus the attenuation of *h* using 3GPP-LTE channel model EVA.

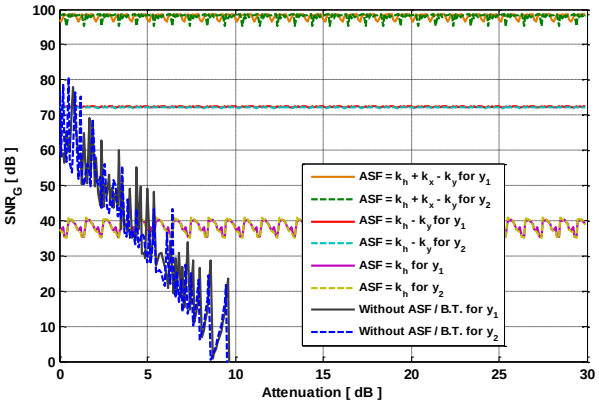

Figure 13. ASF impact on global SNR versus the attenuation of *h* using TGn model E

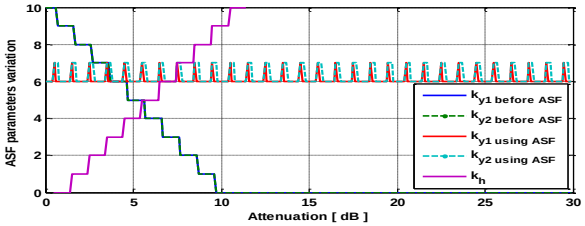

Figure 14. ASF parameters impact versus the attenuation of *h* using TGn model E

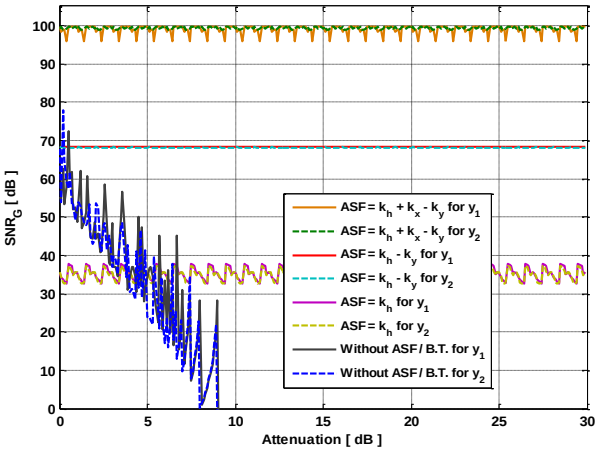

Figure 15. ASF impact on global SNR versus the attenuation of *h* using 3GPP-LTE model EVA

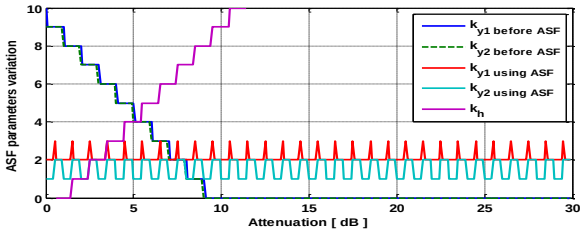

Figure 16. ASF parameters impact versus the attenuation of *h* using 3GPP-LTE model EVA

Using all the coefficient of *ASF*, the output global SNR achieve 100 dB and it remains above 97 dB for high attenuation of h. Moreover, 80 dB is considered as a high accuracy. Thus, the number of bits in the architecture can be decreased to obtain an output global SNR of 80 dB and a lower occupation on the FPGA to simulate higher order MIMO channels. Also, the result shows the benefit of the ST on the BT.

Table 5 presents the new values of the global output SNR before and after using ASF.

Table 5: Global relative error and SNR, for 2×2 MIMO ASF-based time domain architecture

	TGn model E with $x_{WLAN}(t)$		*3GPP-LTE EVA with* $x_{LTE}(t)$	
	y_1	y_2	y_1	y_2
Global Relative Error (%)				
With ST	0.0334	0.0328	0.0362	0.0382
With BT	3.9758	3.9435	2.9263	4.1348
With ASF	0.0011	0.0011	0.0011	0.0011
Global Relative SNR (dB)				
With ST	69.52	69.68	68.82	68.35
With BT	28.01	28.09	30.67	27.68
With ASF	99.03	99.17	98.55	98.84

We notice that after adding ASF, the global output SNR increases significantly.

5. Conclusion

In this paper, the input signals parameters and the relative power of the impulse responses has been related to the relative error and SNR of the output signals. After analyzing the influence of these parameters on the output error and SNR, an improvement algorithm based on an Auto-Scale Factor (ASF) has been proposed and analyzed in details. In the context of mobile radio, the input signal and the impulse responses cannot be predicted and they can undergo fading and be strongly attenuated. Thus, the error of the output signals of the channel simulator will increase widely. The proposed solution consists on multiplying the input signal and the impulse responses by an ASF that increases the output signals and makes it possible to quantify them on a higher number of bits in order to decrease the error at the output. Moreover, the received signal is divided by the correct ASF to obtain the correct output. The new ASF-based architecture has been presented, designed and tested.

For our future work, simulations made using a Virtex-VII [33] XC7V2000T platform will allow us to simulate up to 300 SISO channels. In parallel, measurement campaigns will be carried out with the MIMO channel sounder realized by IETR to obtain the impulse responses of the channel for various types of environments. The final objective of these measurements is to obtain realistic MIMO channel models in order to supply the hardware simulator. A graphical user interface will also be designed to allow the user to reconfigure the simulator parameters.

6. Acknowledgements

This work is a part of CEDRE program and PALMYRE-II project with the support of “Region Bretagne”.

REFERENCES

[1] A. A. Gaston and W. H. Chriss, “A Multipath fading simulator for mobile radio”, *IEEE Transaction on Vehicular Technology*, vol. 22, pp. 241-244, 1973.

[2] R. Fitting, “Wideband troposcatter radio channel simulator”, *IEEE Transaction Communication Technology*, vol. 15, pp. 565-570, 1975.

[3] J. R. Ball, “A real-time fading simulator for mobile radio”, *Radio Electron. Eng.*, vol. 52, pp. 475-478 , 1982.

[4] M. Lecours and F. Marceau, “Design and implementation of channel simulator for wideband mobile radio transmission”, *IEEE VTC*, CA, USA, 1989.

[5] R. A. Comroe and F. Marceau, “All-digital fading simulator”, *Nut. Electron. Conf.*, 1978.

[6] R. A. Goubran, H. M. Hafez and A. U. Sheikh, “Real-time programmable land mobile channel simulator”, *IEEE VTC*, CA, USA, 1986.

[7] J. F. An, A. M. Turkmani and J. D. Parson, “Implementation of a DSP-based frequency non-selective fading simulator”, *Fifth Int. Conf. Radio Receivers Assoc. Syst.*, 1990.

[8] P. J. Cullen, P. C. Fannin and A. Garvey, “Real-time simulation of randomly time-variant linear systems: the mobile radio channel”, *IEEE Trans. Instrum. Meas*., vol. 43, pp. 583-591, 1994.

[9] A. K. Salkintzis, “Implementation of a digital wide-band mobile channel simulator”, *IEEE Transaction on Broadcasting*, vol. 45, pp. 122-128, 1999.

[10] J. R. Papenfuss and M. A. Wickert, “Implementation of a real-time, frequency selective, RF channel simulator using a hybrid DSP-FPGA architecture”, *IEEE Radio Wireless Conf.*, 2000.

[11] S. Fischer, R. Seeger and K. D. Kammeyer, “Implementation of a real-time satellite channel simulator for laboratory and teaching purposes”, *The Third European DSP Education & Research Conference*, 2000.

[12] C. Komninakis, “A Fast and accurate rayleigh fading simulator”, *IEEE GLOBECOM*, Chicago, USA, 2003.

[13] M. Khars and C. Zimmer, “Digital signal processing in a real time propagation simulator”, *IEEE Transaction on Instrum. Meas.*, vol. 55, pp. 197-205, 2006.

[14] S. Kandeepan and A. D. Jayalath, "Narrow-band channel simulator based on statistical models implemented on Texas instruments C6713 DSP and national instruments PCIE-6259 hardware", *ICCS*, Singapore, 2006.

[15] P. Murphy, F. Lou, A. Sabharwal, P. Frantz, "An FPGA Based Rapid Prototyping Platform for MIMO Systems", *Asilomar Conf. on Signals, Systems and Computers, ACSSC*, Vol. 1, 9-12 Nov. 2003, pp. 900-904.

[16] P. Murphy, F. Lou, A. Sabharwal and P. Frantz, "An FPGA Based Rapid Prototyping Platform for MIMO Systems", *ACSSC*, 2003.

[17] S. Picol, G. Zaharia, D. Houzet and G. El Zein, "Hardware simulator for MIMO radio channels: Design and features of the digital block", *IEEE VTC Fall*, Calgary, Canada, 2008.

[18] F. Carames, M. Gonzalez-Lopez and L. Castedo, "FPGA-based vehicular channel emulator for evaluation of IEEE 802.11p transceivers", *Intelligent Transport Systems Telecommunications (ITST)*, 2009.

[19] K. C. Borries, G. Judd, D. D. Stancil and P. Steenkiste, "FPGA-Based channel simulator for a wireless network emulator", *IEEE VTC*, 2009.

[20] H. Eslami, S. V. Tran and A. M. Eltawil, "Design and implementation of a scalable channel Emulator for wideband MIMO systems", *IEEE Transaction on Vehicular Technology*, vol. 58, no. 9, pp. 4698-4708, 2009.

[21] S. Fouladi Fard, A. Alimohammad, B. Cockburn and C. Schlegel, "A single FPGA filter-based multipath fading emulator", *IEEE GLOBECOM*, Honolulu, 2009.

[22] S. Buscemi and R. Sass, "Design of a scalable digital Wireless Channel Emulator for networking radios", *Military Communications Conference (MILCOM)*, Charleston, SC, USA, 2011.

[23] M. I. Akram and A. U. Sheikh, "Design and implementation of real time wideband channel simulator", *EURASIP Journal on Wireless Communications and Networking*, 2012.

[24] "Wireless Channel Emulator", Spirent Communications, 2006.

[25] "Baseband Fading Simulator ABFS, Reduced costs through baseband simulation", Rohde & Schwarz, 1999.

[26] S. Picol, G. Zaharia, D. Houzet, G. El Zein, "Design of the digital block of a hardware simulator for MIMO radio channels", *IEEE PIMRC*, Helsinki, Finland, Sep. 2006.

[27] V. Erceg, L. Shumacher, P. Kyritsi, et al., "TGn Channel Models", IEEE 802.11- 03/940r4, May 10, 2004.

[28] Agilent Technologies, "Advanced design system – LTE channel model - R4-070872 3GPP TR 36.803 v0.3.0", 2008.

[29] H. Farhat, R. Cosquer, G. Grunfelder, L. Le Coq, G. El Zein, "A dual band MIMO channel sounder at 2.2 and 3.5 GHz", *Instrumentation and Measurement Technology Conference Proceedings*, Victoria, Canada, May 2008.

[30] P. Almers, E. Bonek et al., "Survery of Channel and Radio Propagation Models for Wireless MIMO Systems", *EURASIP Journal on Wireless Communications and Networking*, Article ID 19070, 2007.

[31] J. Salz, J.H. Winters, "Effect of fading correlation on adaptive arrays in digital mobile radio", *IEEE Trans. on Veh. Technol.*, Vol. 43, No. 4, AT&T Bell Labs, Holmdel, NJ, Nov. 1994, pp. 1049-1057.

[32] L. Schumacher, K. I. Pedersen, P.E. Mogensen, "From antenna spacings to theoretical capacities – guidelines for simulating MIMO systems", *in Proc. PIMRC Conf.*, Vol. 2, Sept. 2002, pp. 587-592.

[33] "Xilinx: FPGA, CPLD and EPP solutions", www.xilinx.com.

[34] B. Habib, G. Zaharia, G. El Zein, "MIMO hardware simulator: digital block design for 802.11ac applications with TGn channel model test", *IEEE VTC Spring*, Yokohama, Japan, May 2012.

[35] B. Habib, G. Zaharia, G. El Zein, "Digital block design of MIMO hardware simulator for LTE applications", *IEEE ICC*, Ottawa, Canada, June 2012.

[36] D. Umansky, M. Patzold, "Design of Measurement-Based Stochastic Wideband MIMO Channel Simulators", *IEEE Globecom*, Honolulu, Nov. 2009.

[37] M. Al Mahdi Eshtawie, M. Bin Othma, "An algorithm proposed for FIR Filter coefficients representation", *World Academy of Science, Engineering and Technology*, 2007.

[38] B. Habib, G. Zaharia, G. El Zein, "MIMO Hardware Simulator: New Digital Block Design in Frequency Domain for Streaming Signals", *Journal of Wireless Networking and Communications*, Vol. 2, No. 4, pp. 55-65, doi: 10.5923/j.jwnc.20120204.05, 2012.

[39] W. C. Jakes, "Microwave mobile communications", *Wiley & Sons*, New York, Feb. 1975.

[40] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation", *IEEE Journal on Selected Areas of Commun.*, Vol. 20, No. 6, Aug. 2002, pp. 1211-1226.

[41] Q. H. Spencer, et al., "Modeling the statistical time and angle of arrival characteristics of an indoor environment", *IEEE J. Select. Areas Commun.*, Vol. 18, No. 3, March 2000, pp. 347-360.

[42] C-C. Chong, D. I. Laurenson, S. McLaughlin, "Statistical Characterization of the 5.2 GHz wideband directional indoor propagation channels with clustering and correlation properties", *in proc. IEEE Veh. Technol. Conf.*, Vol. 1, Sept. 2002, pp. 629-633.

[43] "ModelSim - Advanced Simulation and Debugging", http://model.com.